3D edge detection using recursive filtering : application to scanner images

Olivier Monga, Rachid Deriche

INRIA

UNITE DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N°930

Programme 6

3D EDGE DETECTION USING RECURSIVE FILTERING: APPLICATION TO SCANNER IMAGES

Olivier MONGA
Rachid DERICHE

Novembre 1988

3D Edge Detection Using Recursive Filtering: Application to Scanner Images

Olivier MONGA
INRIA - Domaine de Voluceau - B.P. 105
78153 LE CHESNAY CEDEX - FRANCE

Rachid DERICHE
INRIA Sophia-Antipolis - 2004, Route des Lucioles
06565 VALBONNE CEDEX - FRANCE

Abstract

This paper deals with the presentation of a new algorithm for three dimensional edge detection. The described method is an extension to the 3D case of the optimal 2D edge detector recently introduced by R. Deriche [2]. It presents better theoretical and experimental performances than some classical approachs used at this date. Some experimental results obtained on magnetic resonance images and on echographic images are shown.

Détection de contours 3D par filtrage récursif : application aux images médicales

Résumé

On propose un nouvel algorithme de détection de contours dans des images tridimensionnelles. Cette méthode est une extension du cas 3D de l'opérateur optimal 2D introduit récemment par R. Deriche. Nous obtenons des performances théoriques et expérimentales meilleures que celles des approches classiques. De plus notre approche se généralise à un nombre de dimensions quelconques ((x, y, z, t) par exemple). Nous présentons des résultats expérimentaux sur des images de résonnance magnétique nucléaire ainsi que sur des images d'échographie.

Cette approche permet de traiter de manière unifiée le problème de la détection de contours dans des images de dimension quelconque.

1 Introduction

In many imaging applications, particularly medicine or industry, the images are three dimensional. Modern scanning techniques, such as computed tomography provide three dimensional images of internal structure of the body [3]. In CT images the luminance function is proportional to the density. Therefore the volumes which present a homogeneous grey level distribution correspond to the various entities of the 3D structure.

A first stage toward the identification and the modelisation of these volumes is to search the surfaces which form their boundaries. The points of such surfaces are located where the gradient of the grey level function is high. This leads to the classical problem of edge detection widely investigated for 2D images [8], but which has not yet received much attention in 3D. Basically 2D or 3D edge detection set the same problem i.e. how to detect the discontinuities of a discrete (2D or 3D) function.

In a first approach Lui [5] proposed a generalization in 3D of the Roberts operator using a symmetric gradient operator [5]. Zucker and Hummel proposed an extension in 3D of the Hueckel operator [4]. An edge is considered as a plane passing through the center of a sphere. Their purpose is to define this plane in an optimal way. With an approach adopted by Hueckel [4] they formulate this problem as one of functional analysis and derive an optimal operator. Their optimal operator turns out to be a generalization of the 2D Sobel operator. Morgenthaler and Rosenfeld [8] propose an edge detection by fitting surfaces plane, quadric to a neighborhood of each image point and by taking the magnitude of the gradient of the surface as an estimate of the rate of change of grey level at that point. Their approach is applicable to multidimensional arrays of data and of course for 3D images. O. Monga and R. Deriche [6] use a 3D version of the Canny's detector. The existing 3D edge detectors [5] - [6] are issued from a generalization of 2D edge detectors.

However the huge size of 3D data makes the algorithmic complexity and the storage requirement be central concerns in the investigation of 3D edge finding.

Particularly the size of the convolution masks used to implement the operator cannot be too large in order to avoid very prohibitive computing times. The convolution of a 3D image $\dim x \times \dim y \times \dim z$ by a mask $n \times n \times n$ costs $\sigma(n^3 \times \dim x \times \dim y \times \dim z)$. For example, this complexity makes almost impossible to use sizes of masks greater than $5 \times 5 \times 5$ for an image $256 \times 256 \times 64$. Therefore an important limitation of many 3D edge detectors has been the use of small convolution masks very sensitive to noise.

Recently, using Canny's design [1] R. Deriche [2] proposed an optimal operator implemented by recursive filtering. In addition to the noise immunity which results, the recursive nature of the filtering operations leads to a substantial saving in computational effort (12 multiplications and 10 additions for one pixel, independently of the size of the convolution masks).

The method presented in this paper is a direct generalization to the 3D case of the Deriche's edge detector. The presentation proceeds as follows: Section 2 is devoted to Deriche's approach. In section 3 we describe our 3D edge detector. Section 4 presents results we have obtained on 3D magnetic resonance images of heart. Section 5 deals with the theoretical and experimental comparison of our algorithm with some existing methods.

In section 6, we illustrate theoretically and experimentally the reasons why to perform 3D edge detection instead of 2D edge detection. In section 7 we investigate how to use our method in order to detect edges in a 2D temporal sequence of echographic images.

2 Deriche's approach

Deriche has proposed an optimal edge operator with respect to Canny's criteria and which can be recursively implemented [2].

In his SM thesis, Canny [1] has made noise a central concern in his investigation of edge finding. He first consider the one–dimensional case edge detection problem with the traditional model of a step in white gaussian noise. He formulates precisely the criteria for effective edge detection: good detection, good localization, one response to one edge. Using the calculus of variations, he derived a general solution for an optimal operator of width W:

$$\begin{aligned} f(x) &= a_1 \times e^{\alpha x} \times \sin(\omega \times x) + a_2 \times e^{\alpha x} \times \cos(\omega \times x) \\ &\quad + a_3 \times e^{-\alpha x} \times (\sin\omega \times x) + a_4 \times e^{-\alpha x} \times (\cos\omega \times x) + C \end{aligned} \tag{1}$$

α, ω, C positive reals.

subject to the boundary conditions:

$$f(0) = 0 \qquad f(W) = 0 \qquad f'(0) = S \qquad f'(W) = 0 \tag{2}$$

On inspection of the shape of this optimal operator, he observed that it is approximately the first derivative of a gaussian:

$$g(x) = -\frac{x}{\sigma^2} \times e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

Deriche points out that the Canny's design was developed for finite extent antisymmetric filter and that the boundary conditions given by 2 becomes in the case of infinite filters:

$$f(0) = 0 \qquad f(+\infty) = 0 \qquad f'(0) = S \qquad f'(+\infty) = 0 \tag{4}$$

Then he derives a solution to (1):

$$h(x) = -c \times e^{-\alpha|x|} \times \sin(\omega \times x) \tag{5}$$

He shows that the Canny's performance criteria are higher for this operator than for the first derivative of a Gaussian. The parameter α is adjusted to yield the desired localization or signal to noise ratio.

He notices that the best trade–off: localization–detection is when $\omega \simeq 0$ i.e.:

$$\sin(\omega x) \simeq \omega x$$

$$d(x) = -c \times x \times e^{-\alpha|x|} \tag{6}$$

He derived a recursive realization of the optimal filter $d(x)$ by two recursive filters moving in opposite directions.

$$y^+(m) = x(m-1) - b_1 \times y^+(m-1) - b_2 \times y^+(m-2) \quad \text{for } m = 1, \cdots, M \tag{7}$$

$$y^-(m) = x(m+1) - b_1 \times y^-(m+1) - b_2 \times y^-(m+2) \quad \text{for } m = M, \cdots, 1 \tag{8}$$

$$y(m) = a(y^+(m) - y^-(m)) \quad \text{for } m = 1, \cdots, M \tag{9}$$

$x(m)$ and $y(m)$ are respectively the input and the output signals

$$\begin{aligned} a &= -c \times e^{-\alpha} \\ b_1 &= -2 \times e^{-\alpha} \\ b_2 &= e^{-2\alpha} \end{aligned} \tag{10}$$

He also shows that $l(x)$ integral of $d(x)$ can also be implemented recursively. This is applied to the extension to 2D or 3D case.

$$l(x) = s \times (\alpha \times \mid x \mid +1)e^{-\alpha \times |x|} \tag{11}$$

$$
\begin{aligned}
y^+(m) &= a_0 \times x(m) + a_1 \times x(m-1) - b_1 \times y^+(m-1) - b_2 \times y^+(m-2) \quad (12)\\
\text{for } m &= 1, \cdots, M
\end{aligned}
$$

$$
\begin{aligned}
y^-(m) &= a_2 \times x(m+1) + a_3 \times x(m+2) - b_1 \times y^-(m+1) - b_2 \times y^-(m+2) \quad (13)\\
\text{for } m &= M, \cdots, 1
\end{aligned}
$$

$$
y(m) = y^+(m) + y^-(m-1) \quad (14)
$$

$$
\begin{aligned}
a_0 &= s \ ; \ a_1 = s \times (\alpha - 1)e^{-\alpha} \ ; \ d_2 = a_1 - s \times b_1 \ ; \ a_3 = -s \times b_2 \quad (15)\\
b_1 &= -2 \times e^{-\alpha} \ ; \ b_2 = e^{-2\alpha}\\
s &= \frac{(1 - e^{-\alpha})^2}{(1 + 2\alpha \times e^{-\alpha} - e^{-2\alpha})}
\end{aligned}
$$

These operators has been extended to the 2-D case by making use of the fact that the slope of a surface in any direction can be determined exactly from its slopes in two directions [2]. In the next section we will show how to extend it to the 3-D case.

3 An optimal 3-D operator

3.1 Principle of the algorithm

In one dimension the step edge is defined in space with one position coordinate. In three dimensions an edge has also an orientation in the 3-D space. Suppose we want to detect edges of a particular orientation. We generate a three dimensional mask for this orientation by convolving a linear edge detection function aligned normal to the edge direction with two projection functions aligned with the two directions orthogonal to the one of the edge. In fact we do not have to differentiate normal to every possible edge direction because the tangent plane of a smooth volume in any direction can be determined exactly from its slopes in three directions.

Thus we create for the x-direction (y-direction respectively) (z-direction respectively) a three dimensional separable mask, product of the edge detector aligned with the x-direction (y-direction respectively) (z-direction respectively) with two projection functions parallel to the y-direction and z-direction (x-direction and z-direction respectively) (x-direction and y-direction respectively). The projection function which can be considered as the smoothing function is chosen to be the integral of the edge detector. Using these three masks, we compute at each point the three components of the gradient. In a second

stage we determine the local maxima of the gradient magnitude in the direction of the gradient. This stage is equivalent to detect the zero crossing of the second derivative in the edge direction. In a last stage an hysteresis thresholding allows to remove points with low gradient magnitude.

3.2 Implementation

3.2.1 Computational of the gradient

We use the principles described above with the derivation function $(d(x))$ and the smoothing function $(l(x))$.

Let $I(i,j,k)$ be a three-dimensional image.

The three components of the gradient are performed using the following three convolutions masks $X(i,j,k), Y(i,j,k), Z(i,j,k)$:

$$\begin{aligned} X(i,j,k) &= d(i) \times l(j) \times l(k) \\ &= -c \times i \times e^{-\alpha|i|} \times s \times (\alpha \times \mid j \mid +1) \times e^{-\alpha \times |j|} \times s(\alpha \times \mid k \mid +1) \times e^{-\alpha \times |k|} \\ &= (-c \times i) \times (s \times \alpha \times \mid j \mid +s) \times (s \times \alpha \times \mid k \mid +s) \times e^{-\alpha(|i|+|j|+|k|)} \end{aligned} \tag{16}$$

$$\begin{aligned} Y(i,j,k) = l(i) \times d(j) \times l(k) &= (-c \times j) \times (s \times \alpha \times \mid j \mid +s) \times \\ &\quad (s \times \alpha \times \mid k \mid +s) \times e^{-\alpha(|i|+|j|+|k|)} \end{aligned} \tag{17}$$

$$\begin{aligned} Z(i,j,k) = l(i) \times l(j) \times d(k) &= (-c \times k) \times (s \times \alpha \times \mid j \mid +s) \times \\ &\quad (s \times \alpha \times \mid k \mid +s) \times e^{-\alpha(|i|+|j|+|k|)} \end{aligned} \tag{18}$$

By making use of their separability, the masks $\{X(i,j,k)\}, \{Y(i,j,k)\}$ and $\{Z(i,j,k)\}$ can easily be implemented as a 3-D separable recursive filter.

To convolve a $NLG \times NCL \times NPL$ image $I(i,j,k)$ with the mask $\{X(i,j,k)\}$ we first do a derivative filtering in the x-direction and finally apply the smoothing filter $l(n)$ in the y-direction and z-direction. Using the results reported in section 2 we obtain:

$$\begin{aligned} Y^+(i,j,k) &= I(i-1,j,k) - b_1 \times Y^+(i-1,j,k) - b_2 \times Y^+(i-2,j,k) \\ k &= 1, \cdots, DZ \quad j = 1, \cdots, DY \quad i = 1, \cdots, DX \end{aligned} \tag{19}$$

$$\begin{aligned} Y^-(i,j,k) &= I(i+1,j,k) - b_1 \times Y^-(i+1,j,k) - b_2 \times Y^-(i+2,j,k) \\ k &= 1, \cdots, DZ \quad j = 1, \cdots, DY \quad i = DX, \cdots, 1 \end{aligned} \tag{20}$$

$$\begin{aligned} S(i,j,k) &= a(Y^+(i,j,k) - Y^+(i,j,k)) \\ k &= 1,\cdots,DZ \quad j = 1,\cdots,DY \quad i = 1,\cdots,DX \end{aligned} \tag{21}$$

$$\begin{aligned} R^+(i,j,k) &= a_0 \times S(i,j,k) + a_1 \times S(i,j-1,k) \\ &\quad -b_1 \times R^+(i,j-1,k) - b_2 \times R^+(i,j-2,k) \\ k &= 1,\cdots,DZ \quad i = 1,\cdots,DX \quad j = 1,\cdots,DY \end{aligned} \tag{22}$$

$$\begin{aligned} R^-(i,j,k) &= a_2 \times S(i,j+1,k) + a_3 \times S(i,j+2,k) \\ &\quad -b_1 \times R^-(i,j+1,k) - b_2 \times R^-(i,j+2,k) \\ k &= 1,\cdots,DZ \quad i = 1,\cdots,DX \quad j = DY,\cdots,1 \end{aligned} \tag{23}$$

$$\begin{aligned} R(i,j,k) &= R^-(i,j,k) + R^+(i,j,k) \\ i &= 1,\cdots,DX \quad j = 1,\cdots,DY \quad k = 1,\cdots,DZ \end{aligned} \tag{24}$$

$$\begin{aligned} T^+(i,j,k) &= a_0 \times R(i,j,k) + a_1 \times R(i,j,k-1) \\ &\quad -b_1 \times T^+(i,j,k-1) - b_2 \times T^+(i,j,k-2) \\ i &= 1,\cdots,DX \quad j = 1,\cdots,DY \quad k = 1,\cdots,DZ \end{aligned} \tag{25}$$

$$\begin{aligned} T^-(i,j,k) &= a_2 \times R(i,j,k+1) + a_3 \times R(i,j,k+2) \\ &\quad -b_1 \times T^-(i,j,k+1) - b_2 \times T^-(i,j,k+2) \\ i &= 1,\cdots,DX \quad j = 1,\cdots,DY \quad k = DZ,\cdots,1 \end{aligned} \tag{26}$$

$$\begin{aligned} T(i,j,k) &= T^-(i,j,k) + T^-(i,j,k) \\ i &= 1,\cdots,DX \quad j = 1,\cdots,DY \quad k = 1,\cdots,DZ \end{aligned} \tag{27}$$

The values of $a, a_0, a_1, a_2, a_3, b_1$ and b_2 are given by the equations 10 and 15.

This recursive implementation allows to convolve the input image with the mask $\{X(i,j,k)\}$ with a computational effort of 20 multiplications and 18 additions per point and without any truncation of the mask. The same scheme can be applied to the masks $\{Y(i,j,k)\}$ and $\{Z(i,j,k)\}$. Therefore the complete computing of the gradient at each point costs: $60 \times DX \times DY \times DZ$ multiplications and $54 \times DX \times DY \times DZ$ additions.

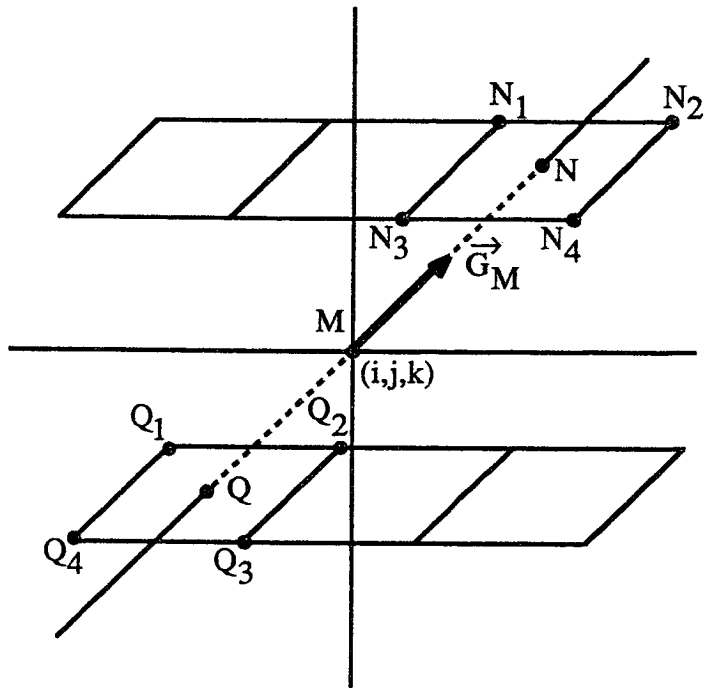

Figure 1 :

3.2.2 Extraction of the local gradient extrema

We now describe how we extract the local maxima of the magnitude of the gradient in the direction of the gradient.

Let $GX(i,j,k), GY(i,j,k)$ and $GZ(i,j,k)$ be respectively the gradient images in the directions X, Y and Z. At each point is attached the 24 squared facets of the cube defined by its neighbors. Each facet is a square defined by 4 points. The line which contains the current point and which is parallel to the gradient at that point intersects 2 facets. A linear interpolation is performed in order to find to gradient at the two intersection points (for decreasing the sensibility of our method to the noise). This interpolation uses the gradient at the four points of the facets. The points which are local maxima for the gradient's magnitude are selected. In practice, to avoid double edge points, we choose the points which are strict maxima in the sense of gradient and large maxima in the other sense. This is equivalent to select edge points in the region of higher intensity or in the region of lower intensity. Figure 1 illustrates our scheme:

$M(i,j,k)$ is a point whose gradient is $\vec{G}_M$.

The line whose direction is $\vec{G}_M$ intersects the two squared faces $N_1N_2N_3N_4$ and $Q_1Q_2Q_3Q_4$ at points N and Q.

We approximate the vector gradients at points N and Q respectively $\vec{G}_N$ and $\vec{G}_Q$ as:

$$\begin{aligned}\vec{G}_N &= \frac{(d(N,N_1)\times\vec{G}_{N_2}+d(N_3,N)\times\vec{G}_{N_3}+d(N_2,N)\times\vec{G}_{N_2}+d(N_4,N)\times\vec{G}_{N_4})}{(d(N,N_1)+d(N,N_2)+d(N,N_3)+d(N,N_4))}\\ \vec{G}_Q &= \frac{(d(Q,Q_1)\times\vec{G}_{Q_1}+d(Q,Q_2)\times\vec{G}_{Q_2}+d(Q,Q_3)\times\vec{G}_{Q_3}+d(Q,Q_4)\times\vec{G}_{Q_4})}{(d(Q,Q_1)+d(Q,Q_2)+d(Q,Q_3)+d(Q,Q_4))}\end{aligned}$$

The point M is selected if $\|\vec{G}_M\| \geq \|\vec{G}_N\|$ and $\|\vec{G}_M\| > \|\vec{G}_Q\|$.

3.2.3 Hysteresis thresholding

In a last stage, we eliminate the points where the gradient's magnitude is low by a thresholding with hysteresis [1]. We realize an expansion in 3-D connected components from all local extrema whose gradient magnitude is greater than a high threshold HT to the local extrema presenting a magnitude greater than a low threshold LT. This thresholding strategy is particularly efficient in the 3-D case because it enables to get good connected edge points. This is of great interest for regrouping these points in order to built surfaces.

4 Results

We have tested our method on 3-D magnetic resonance images of heart given by Hôpital Kremlin Bicêtre, Paris. A cardiac cycle has been divided in eight times. For each time 46 parallel cross sections of size 256×256 are computed. Therefore we obtain eight 3-D image of size: $256 \times 256 \times 46$ corresponding respectively to the eight cardiac times. The resolution is the same along the three directions (1.5 mm).

We consider that the noise has the same characteristic in the three directions and the we use a same operator width α (see 2 – 3) to smooth or to derive in any direction. We have experimented several values for α and $\alpha = 1$ seems to be, on these images, a good trade–off between detection and localization. Figures 5–7, and 19–24 show some results we have obtained. We notice the good noise immunity provided by our algorithm, particularly in the areas where the variations of the flow blood degrades hardly the image (this degradation is specific to magnetic resonance images). We also remark that the boundaries of the left ventricle are well defined in spite of the noise present in this region. Our implementation run on a Gould and the complete edge detection scheme takes about 30' CPU for a $256\times256\times46$ image.

These results illustrate the robustness of our 3-D edge detection algorithm.

5 Comparison with previous works

In this section we will compare theoretically and experimentally our 3-D edge detection algorithm with some existing methods.

5.1 How to compare theoretically the performance of two operators

For comparison purposes of two edge detector operators we use the scheme introduced by Canny [1].

In his SM thesis Canny considered the one dimensional case edge detection problem with the traditional model of a step in white Gaussian noise.

Let the amplitude of the step be A, and let the variance of the input white noise be n_0^2, The input signal $I(x)$ can be represented by the step:

$$I(x) = A \times u_{-1}(x) + n(x)$$

with $n_0^2 =< n^2(x) >$ for all x.

He assumed that detection is performed by convolving the noisy edge with a spatial antisymmetric function $f(x)$ and marking edges at the maxima in the output $\theta(x_0)$ of this convolution:

$$\theta(x_0) = \int_{-\infty}^{+\infty} I(x) \times f(x_0 - x)dx$$

He formulate precisely the criteria for edge detection:

- Detection criterion: This criterion corresponds to maximize signal to noise ratio SNR which is defined as the quotient of the response to the step only and the square root of the mean squared noise response:

$$SNR = \frac{A}{n_0} \times \frac{\int_{-\infty}^{0} f(x)dx}{(\int_{-\infty}^{+\infty} f^2(x)dx)^{\frac{1}{2}}} = \frac{A}{n_0} \times \Sigma$$

- Localization criterion: The points marked as edges by the operator should be as close as possible to the center of the true edge. This corresponds to minimize the variance of the zero–crossing position or to maximize the localization criterion L.

$$L = \frac{A}{n_0} \times \frac{| f'(0) |}{(\int_{-\infty}^{+\infty} f'^2(x)dx)^{\frac{1}{2}}} = \frac{A}{n_0} \times \lambda$$

These two criteria are antagonist [1] and Canny proposed to merge them in a meaningful way: maximize the product $\sum \times \lambda$. This product can therefore be used to estimate the detection–localization a performance of a 1-D edge operator. For higher dimensions (2-D or 3-D) we will consider the 1-D operator obtained by separation of the operator response along the (2 or 3) directions; and we will compute the performance measure of this 1-D operator.

For Deriche's operator [2] we have:

$$\begin{cases} \lambda & = & (2\alpha)^{\frac{1}{2}} \\ \Sigma & = & (\frac{2}{\alpha})^{\frac{1}{2}} \\ \Sigma, \lambda & = & 2 \end{cases}$$

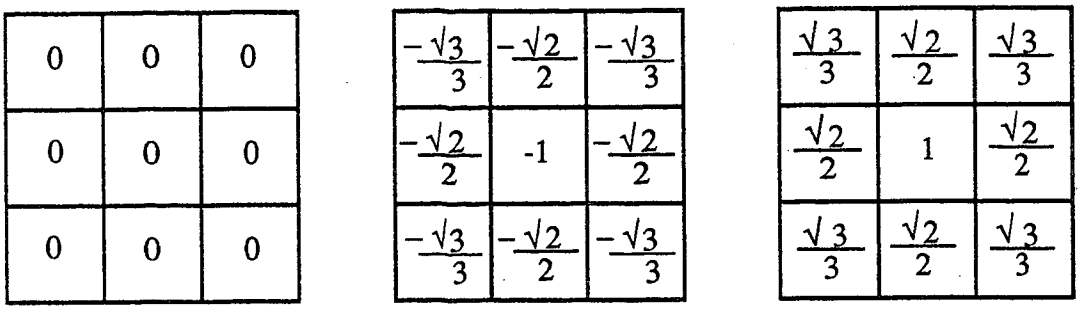

Figure 2 : 3×3 Zucker–Hummel masks

5.1.1 Zucker–Hummel approach

Zucker and Hummel [10] proposes a 3-D edge model which is a direct extension of the 2-D edge model introduced by Hueckel [4]. They consider a 3-D edge as a plane passing through the center of an unit volume. They propose to define this plane in an optimal fashion. They approximate the optimal operator with the three basis functions:

$$\begin{aligned}
\Phi_1(x,y,z) &= \frac{x}{\sqrt{x^2+y^2+z^2}} \\
\Phi_2(x,y,z) &= \frac{y}{\sqrt{x^2+y^2+z^2}} \\
\Phi_3(x,y,z) &= \frac{z}{\sqrt{x^2+y^2+z^2}}
\end{aligned}$$

Φ_1, Φ_2 and Φ_3 corresponds respectively to the three directions X, Y, Z. Two approximations are shown, the first in which the unit sphere is partitioned into a $3 \times 3 \times 3$ unit cube and the second into a $5 \times 5 \times 5$ cube. Theoretically the best operator would be the infinite operator defined by Φ_1, Φ_2, Φ_3.

The Zucker–Hummel operator is not separable in the three directions X, Y, Z then we cannot derive a 1-D equivalent operator. Therefore, in this case the Canny's performance criteria takes no sense. However we notice that the width of the operator corresponds to the size of the sphere which is set to one in [10]. We also remark that the non–separability of the operator makes its implementation very costly for great sizes of masks. Figures 5–9 show results obtained with the 3×3 edge basis functions and with our method on the same data than in the previous paragraph.

We see that the edge localization is roughly the same with the two methods but that our operator is very less sensitive to noise. It should be pointed out that while the computing time required by the Zucker approach is directly proportional to the size of the used mask, the required time for our approach does not change when we deal with large filters (i.e small values for the parameter α).

5.1.2 Monga–Deriche approach

In previous work Monga and Deriche [6] have proposed to use the first derivative of a 3-D gaussian. They obtain three separable masks:

$$\begin{aligned} G_x(x,y,z) &= (\frac{-x}{\sigma^2}) \times e^{\frac{-(x^2+y^2+z^2)}{2\sigma^2}} \\ G_y(x,y,z) &= (\frac{-y}{\sigma^2}) \times e^{\frac{-(x^2+y^2+z^2)}{2\sigma^2}} \\ G_z(x,y,z) &= (\frac{-z}{\sigma^2}) \times e^{\frac{-(x^2+y^2+z^2)}{2\sigma^2}} \end{aligned}$$

The 1-D corresponding operator is:

$$G(x) = \frac{-x}{\sigma^2} \times e^{\frac{-x^2}{2\sigma^2}}$$

Canny [1] has shown that the Canny's performance criteria of this operator are:

$$\Sigma_g \times \lambda_g = 0.92$$

For Deriche's operator we have:

$$\Sigma_d \times \lambda_d = 2$$

Figures 5–7 and 10–11 show results obtained on the same 3-D images than in the precedent paragraph. We have taken σ^2 equal to $\frac{1}{2}$ and α equal to 1 in order to establish a correct comparison. Using large values gor the parameter σ leads to deal a very time consuming approach. This is not the case with the method proposed in this paper.

6 Why to perform 3-D edge detection instead of 2-D edge detection

In this section, we will investigate theoretically and experimental reasons why 3-D edge detection is of greater interest than 2-D edge detection, in CT 3-D images. We point out that similar remarks can be done by comparing 1-D edge detection and 2-D edge detection for 2-D images.

3-D edge detection compared with 2-D edge detection yields:

1. better immunity to the noise
2. better estimation of the gradient magnitude
3. computation of the 3-D gradient

We now discuss and illustrate these three points.

6..3 Better immunity to the noise

We may suppose that noise in CT images is randomly and therefore theoretically not correlated in the three directions.

Our edge detection algorithm consists in smoothing the image along the three orthogonal directions X, Y, Z and then to derive. If we perform only a 2-D edge detection we will only smooth along X, Y and then the noise accidently correlated along X and Y will not be removed. If we perform 3-D edge detection we will also smooth along Z direction and then we will also remove the noise correlated along X and Y and not along Z.

The above remark is illustrated by figures 5–7 and 12–16 which show results provided by 2-D and 3-D edge detection in a cross section where the Z gradient is very low. We notice that in many areas 3-D edge detection yields to remove noisy edge points which are detected by 2-D edge detection. The 2-D edge detection method we used is strictly the 2-D restriction of our 3-D algorithm.

6..4 Better estimation of the gradient magnitude

Let $I(x, y, z)$ be the grey level function of a 3-D image.

Let $\vec{G}(x, y, z)$ be the vector gradient at point : (x, y, z).

$$\vec{G} = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial z} \end{pmatrix}$$

If we have $\frac{\partial I}{\partial z} = 0$ 3-D edge detection yields only to improve the noise immunity. This occurs when the normal to the surface is parallel to the cross section plane. Then the vector gradient $\vec{G}$ is contained in the cutting plane (M, X, Y). Thus in this case the third dimension enables to perform a more efficient smoothing (see above paragraph) but the gradient magnitude remains the same than in 2-D.

If we have $\frac{\partial I}{\partial z} \neq 0$, 3-D edge detection enables to compute more precisely the gradient magnitude. $\frac{\partial I}{\partial z}$ is highest when the normal to the surface is orthogonal to the cross section plane. This is illustrated in figures 5–7 and 19–21. If we do not take into account $\frac{\partial I}{\partial z}$ we obtain a gradient magnitude depending on the relative orientation of the cutting plane (M, X, Y) and of the normal to the surface at point M. The above remark would imply that the next stages: determination of local extrema and thresholding, will be damaged. Figures 23– 26 show examples of differences existing between 2-D and 3-D edge detection when $\frac{\partial I}{\partial z} \neq 0$.

6..5 Computation of the 3-D gradient

Our edge detection method uses the direction and the orientation of the vector gradient to extract the extrema of its magnitude. This stage is improved by considering the z - gradient, particularly in the areas where it is high. This enables to obtain a more precise localization of edge points.

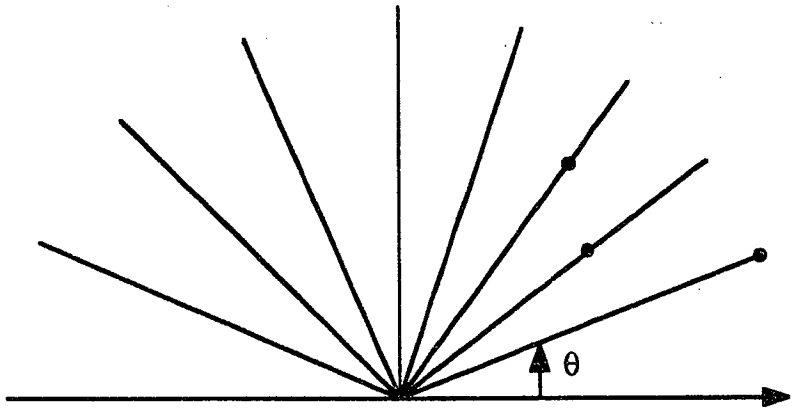

Figure 3 : Extraction of local extrema

Moreover the 3-D gradient at each edge points corresponds to an approximation of the normal to the surface. This is of great interest to modelize surfaces thanks to normal vector fields [9].

7 Application to echographic images

This section shows how to apply our method to sequences of echographic images taken at different times.

We consider echographic images $E_1(r,\theta), E_2(r,\theta), \cdots, E_p(r,\theta)$ such that we can suppose:

1. there exists no discontinuities between two successive images
2. the noise is homogeneous on lines $\theta = cste$ and on circles $r = cste$.

We can define a 3-D spatio-temporal image $E(r,\theta,t)$ from the sequence $(E_i(r,\theta))$. Because of (1) we can suppose that E is continuous along the time axis. Thus the first stage of our edge detection consist on smoothing E on each line $t = cste$ thanks to the smoothing filter (l) defined in 2.

In a second stage we will compute the 2-D gradient on each image $t = cste$. Due to (2) we take interest on determining the first with respect to polar coordinates (r,θ) and then to write it in cartesian coordinates (x,y).

This may be done thanks to the following calculus:

$$
\begin{array}{lcl}
f & : & (x,y) \longrightarrow (r,\theta) = (\sqrt{x^2+y^2}, \arctan\frac{y}{x}) \\
g & : & (r,\theta) \longrightarrow f(r\cos\theta, r\sin\theta) \\
k & = & g \circ f \\
dh_{(x,y)} & = & dg_{(x,y)} \circ df_{(x,y)} = \\
 & & (\frac{\partial g}{\partial r}(r,\theta), \frac{\partial g}{\partial \theta}(r,\theta)) \times \begin{pmatrix} \frac{x}{\sqrt{x^2+y^2}} & \frac{y}{\sqrt{x^2+y^2}} \\ -\frac{y}{\sqrt{x^2+y^2}} & \frac{x}{\sqrt{x^2+y^2}} \end{pmatrix} = \\
 & & (\frac{\partial g}{\partial r}(r,\theta), \frac{\partial g}{\partial \theta}(r,\theta)) \times \begin{pmatrix} \cos\theta & \sin\theta \\ -\frac{\sin}{r} & \frac{\cos\theta}{r} \end{pmatrix} = \\
 & & \begin{pmatrix} \cos\theta \times \frac{\partial g}{\partial r}(r,\theta) & - & \frac{\sin\theta}{r} \times \frac{\partial g}{\partial \theta}(r,\theta) \\ \sin\theta \times \frac{\partial g}{\partial r}(r,\theta) & + & \frac{\cos\theta}{r} \times \frac{\partial g}{\partial \theta}(r,\theta) \end{pmatrix}
\end{array}
$$

it comes:

$$
\begin{cases} \frac{\partial h}{\partial x}(x,y) = \cos\theta \times \frac{\partial g}{\partial r}(r,\theta) - \frac{\sin\theta}{r} \times \frac{\partial g}{\partial \theta}(r,\theta) \\ \frac{\partial h}{\partial y}(x,y) = \sin\theta \times \frac{\partial g}{\partial r}(r,\theta) + \frac{\cos\theta}{r} \times \frac{\partial g}{\partial \theta}(r,\theta) \end{cases} \tag{28}
$$

Therefore to smooth along circular arcs $r = cste(resp.\theta = cste)$ with filter (P) and to derive along line segments $\theta = cste(resp.r = cste)$ provide $\frac{\partial g}{\partial \theta}(resp.\frac{\partial g}{\partial r})$.

Equations 28 allows then to compute the vector gradient in cartesian space. We extract the local gradient extrema and perform an hysteresis thresholding in 2-D space on each image $r = cste$.

Figures 26–27 show results obtained on echographic heart's data.

8 Conclusion

We have presented a new 3-D edge detection algorithm which is an extension of a 2-D edge detection method introduced by Deriche [2].

This method combines better theoretical and experimental performances than the existing ones. Moreover a highly recursive implementation enables a substantial saving in computational effort. We show promising results obtained by this algorithm on RMN and Echographic images. This method is currently tested on scintigraphic nuclear images.

Acknowledgement

The authors would like to thank Dr. J.M. Rocchisani from Hôpital Cochin Paris and Dr. J. Bittoun from Hôpital Kremlin Bicêtre Paris for having provided the RMN images. Thanks are also due to J.S. Jezouin and J.P. Gambotto from Matra for having provided the echographic images.

Références

[1] J.F. Canny. *Finding edges and lines in images.* Technical Report TR. 720, MIT, June 1983.

[2] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2), May 1987.

[3] R. Gordon, G.T. Herman, and S.A. Johnson. Image reconstruction from projections. *Sci. Amer.*, 233:56–68, 1975.

[4] M.T. Hueckel. An operator which locates edges in digitized pictures. *J. Ass. Comput. Mach.*, 18:113–125, 1971.

[5] H.K. Lui. Two and three dimensional boundary detection. In *Comput. Graphics Image Process.*, pages 123–134, 1977. Vol. 6.

[6] O. Monga and R. Deriche. A new three dimensional boundary detection. In *International Conference on Pattern Recognition*, Paris, 1986.

[7] M. Morgenthaler and A. Rosenfeld. Multidimensional edge detection by hypersurface fitting. *PAMI-3*, 4, July 1981.

[8] A. Rosenfeld and A. Kak. Digital image processing. *New York: Academic*, 1976.

[9] P. Sander. *On reliably inferring differential structure from three-dimensional images.* PhD thesis, Mac Gill University, 1988.

[10] S.W. Zucker and R.A. Hummel. A three dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(PAMI-3), May 1981.

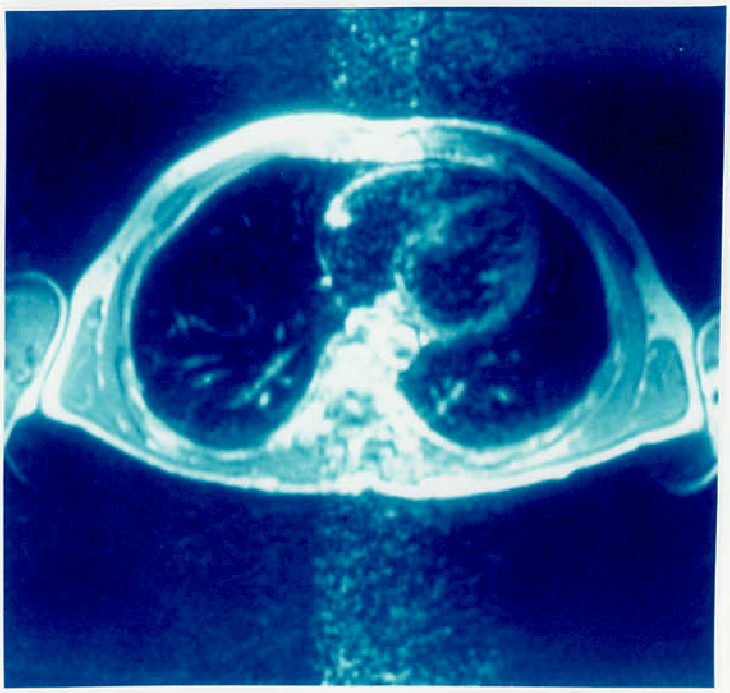

Figure 4: Original cross section of RMN image corresponding to slice 22 at cardiac time 1

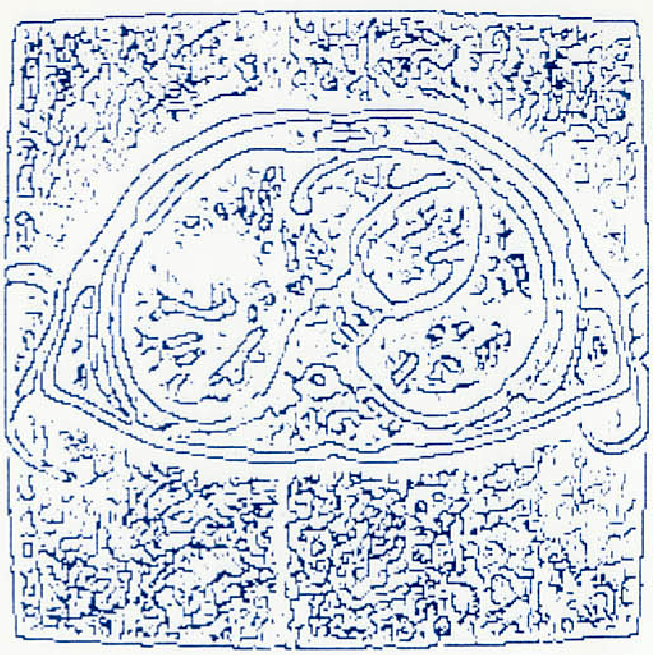

Figure 5: Local 3-D extrema extracted thanks to our method, $\alpha = 1$

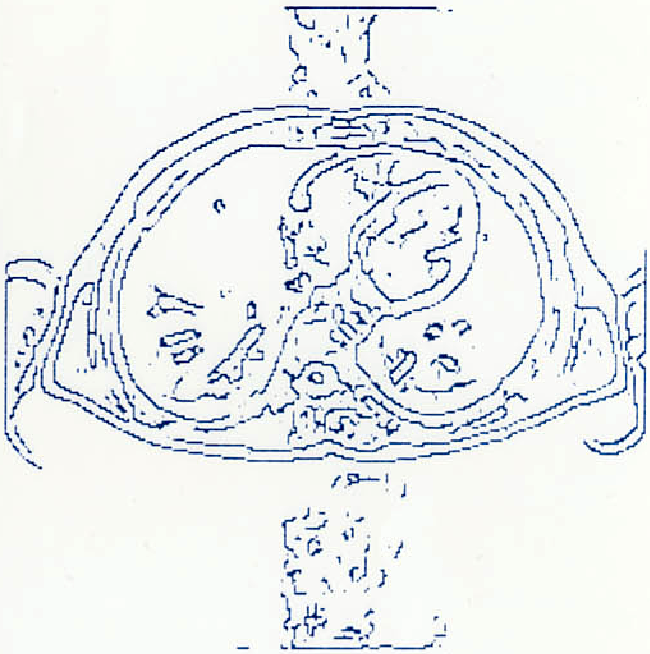

Figure 6: Edges obtained after 3-D hysteresis thresholding of figure 5

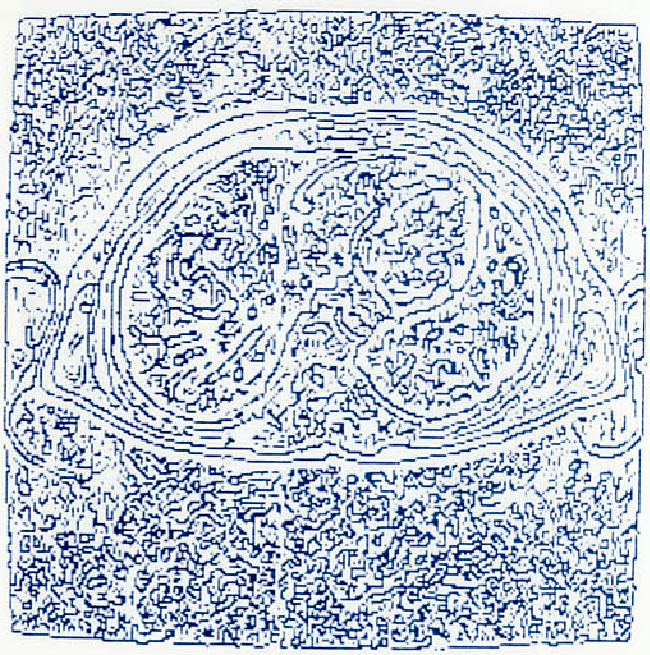

Figure 7: Local 3-D extrema thanks to Zucker–Hummel 3×3 detector corresponding to figure 4

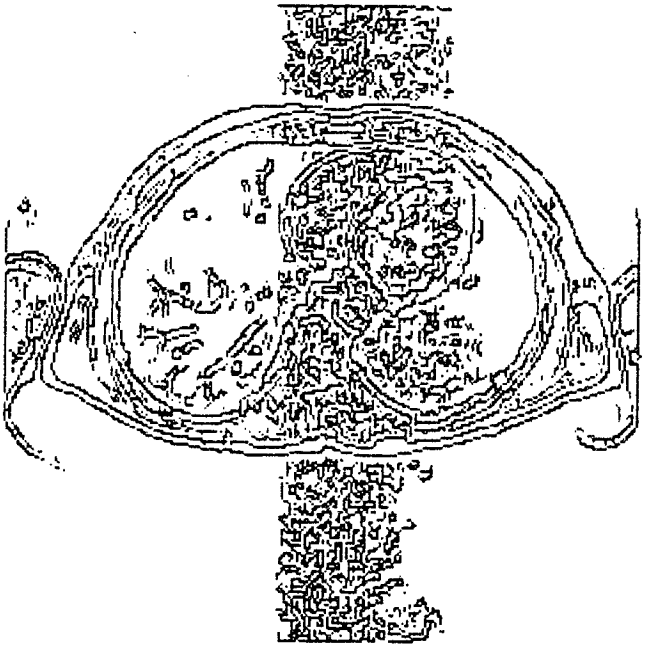

Figure 8: Edges obtained after 3-D hysteresis thresholding of figure 7

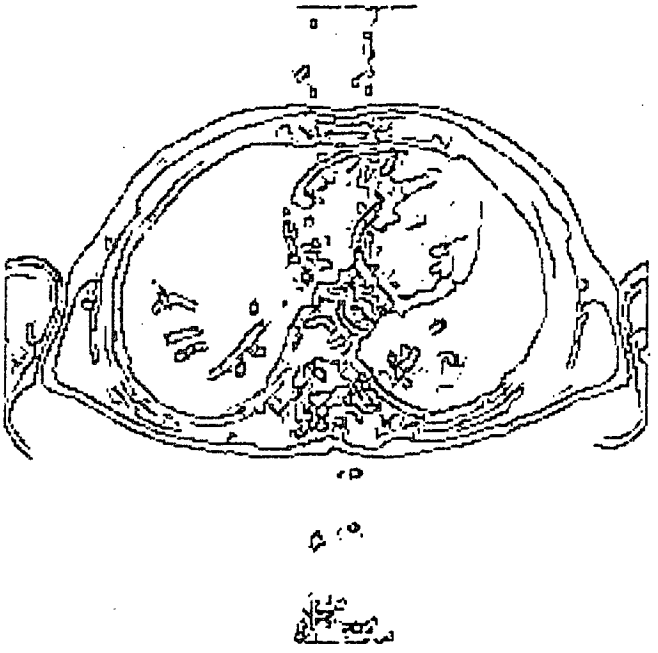

Figure 10: Edges obtained after 3-D hysteresis thresholding of figure 9

Figure 9: Local 3-D extrema extracted thanks to Monga–Deriche detector corresponding to figure 4

Figure 11: 3-D gradient in X-direction corresponding to figure 4

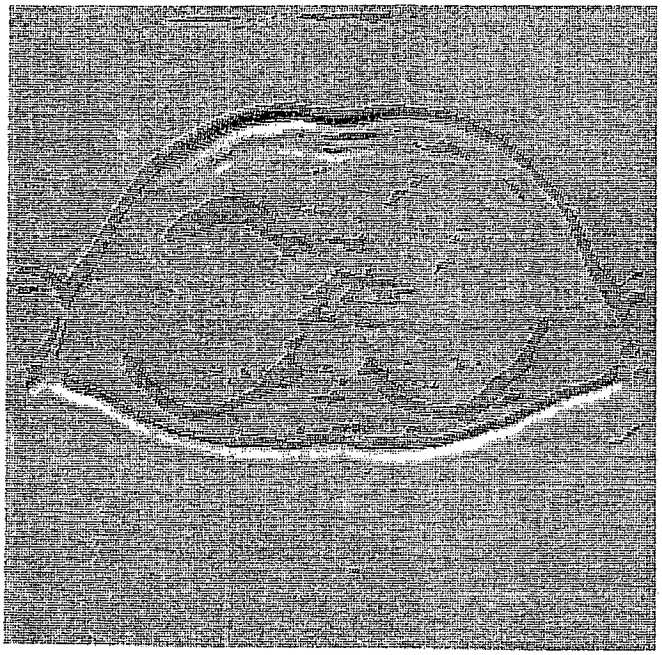

Figure 12: 3-D gradient in Y-direction corresponding to figure 4

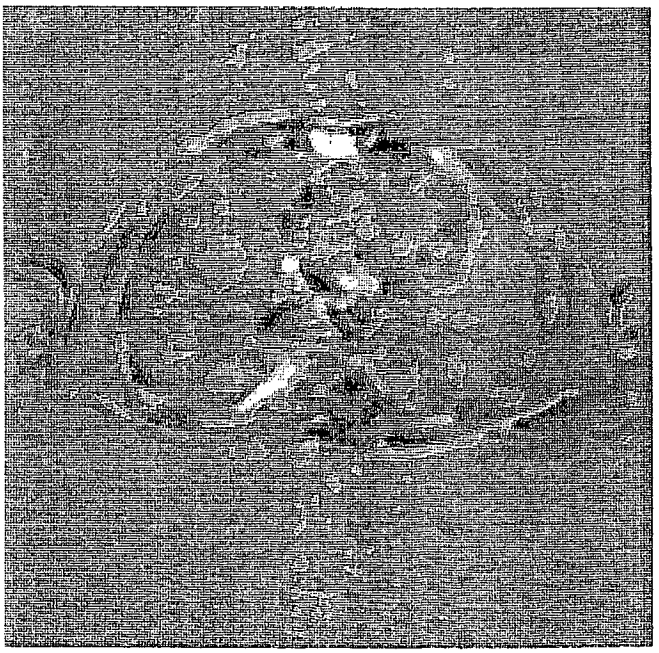

Figure 13: 3-D gradient in Z-direction corresponding to figure 4

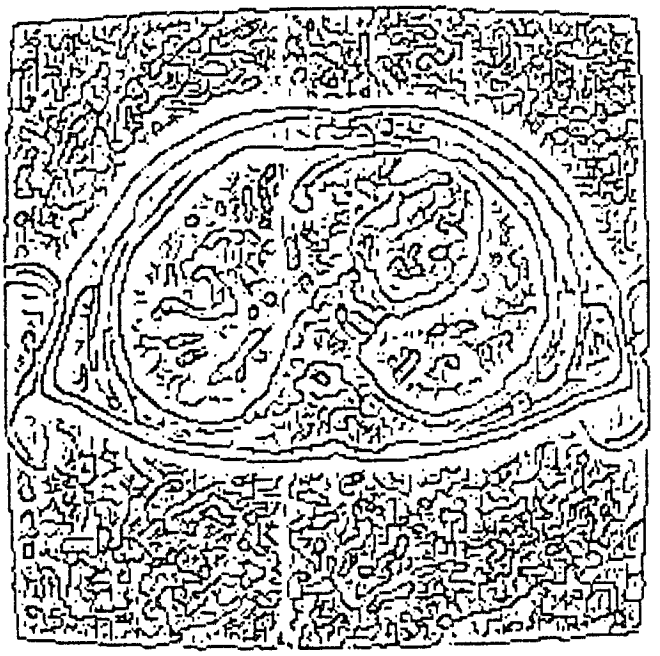

Figure 14: Local extrema extracted thanks to 2-D Deriche detector corresponding to figure 4

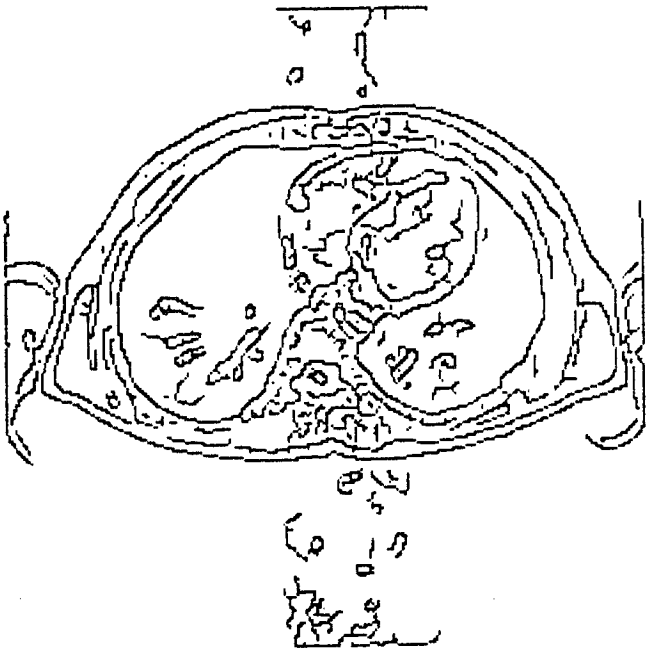

Figure 15: Edges obtained after 2-D hysteresis thresholding of figure 14

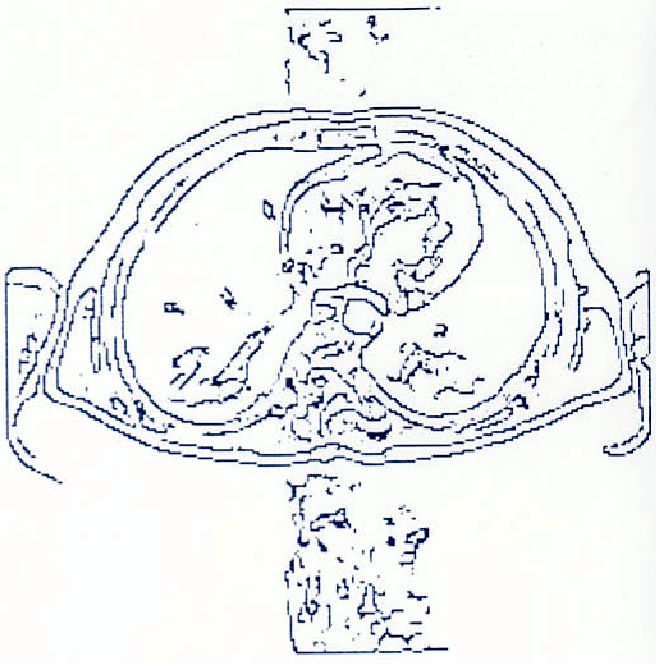

Figure 16: Edges obtained thanks to our method for slice 22 at cardiac time 4

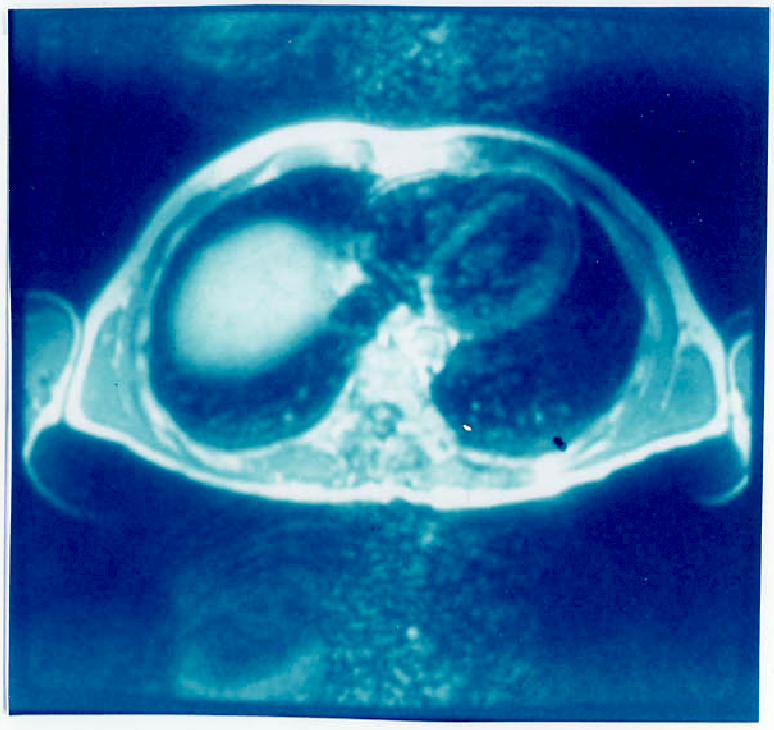

Figure 18: Original cross section corresponding to slice 11 at cardiac time 3

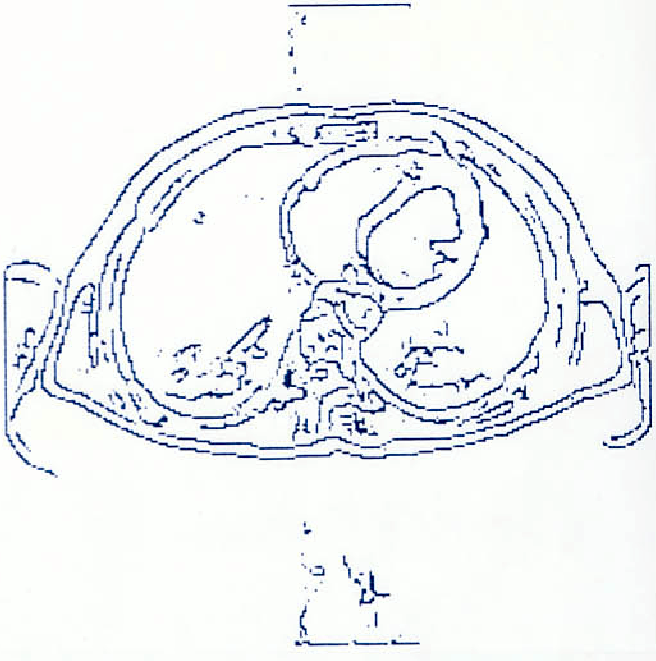

Figure 17: Edges provided by our method for slice 22 at cardiac time 7

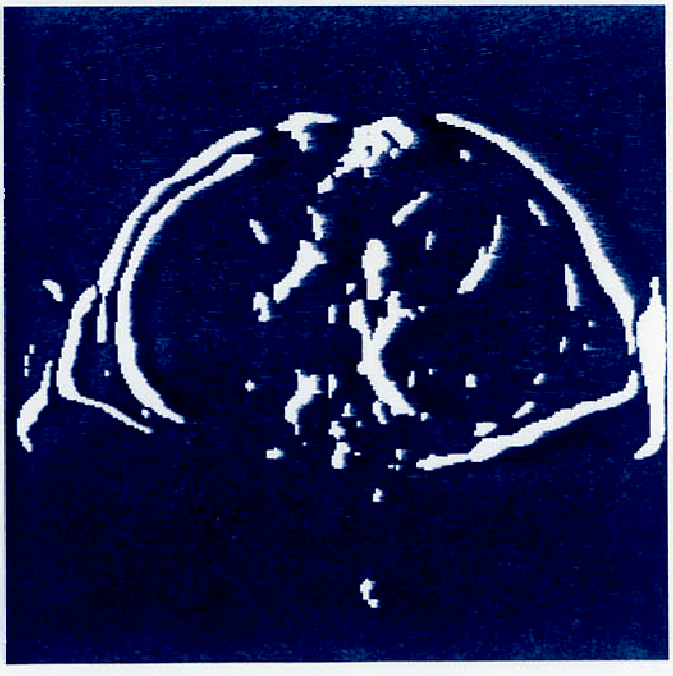

Figure 19: X-gradient corresponding to figure 18

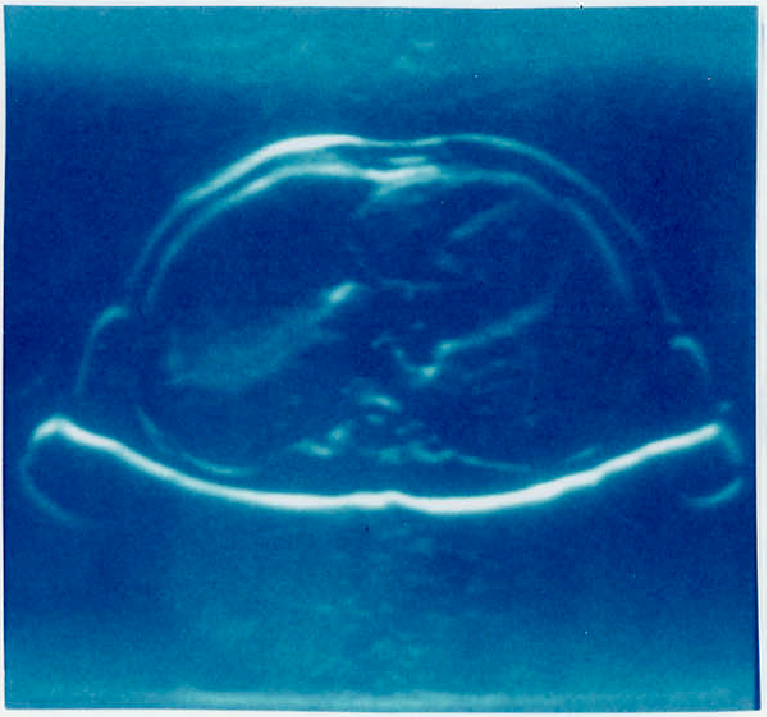

Figure 20: Y-gradient corresponding to figure 18

Figure 21: Z-gradient corresponding to figure 18

Figure 22: 3-D local extrema corresponding to figure 18

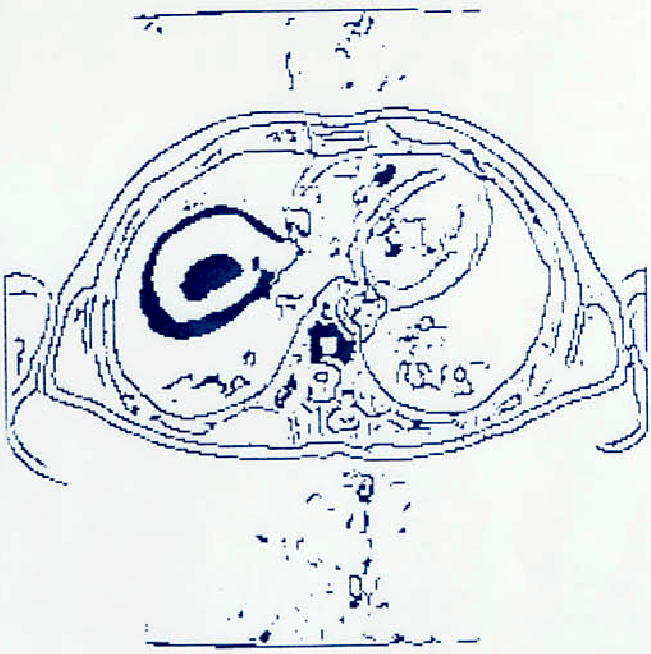

Figure 23: 3-D edges corresponding to figure 18

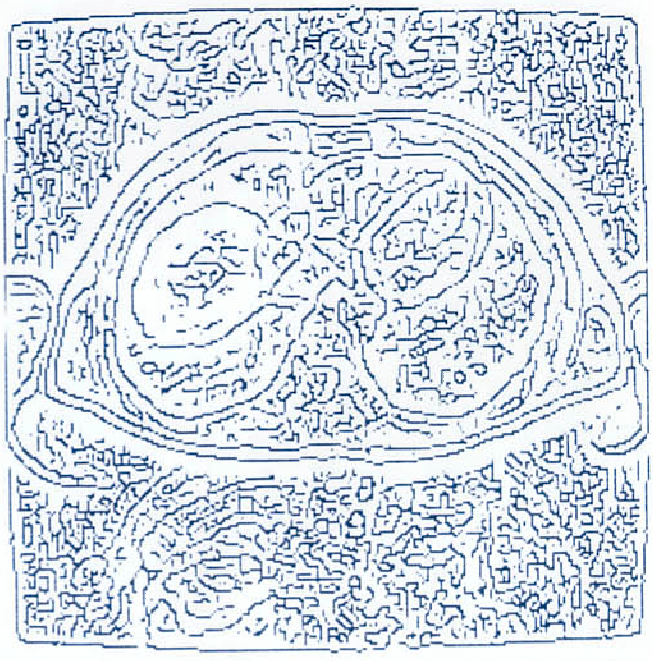

Figure 24: 2-D local extrema corresponding to figure 18

Figure 26: Original echographic image

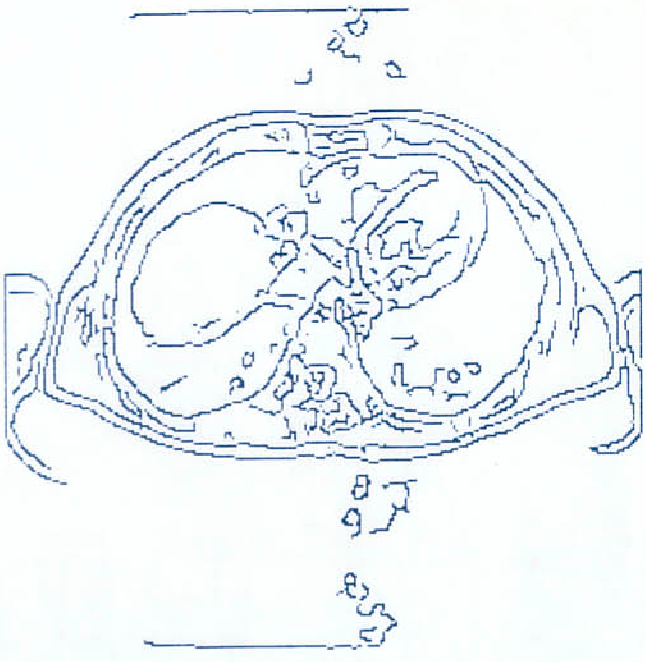

Figure 25: 2-D edges extrema corresponding to figure 18

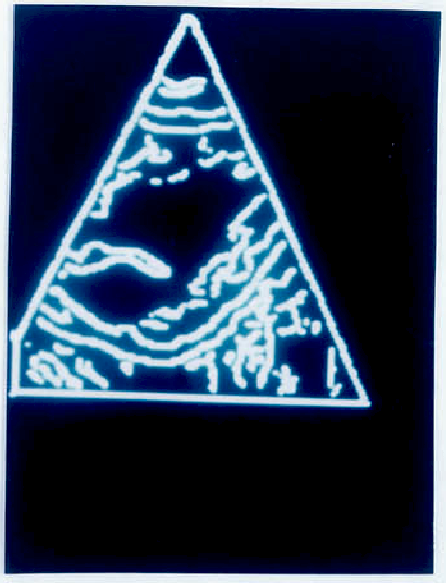

Figure 27: Edges corresponding to figure 26